

0325.00524  
CD01153

**TECHNIQUE FOR PAYLOAD-INDEPENDENT FRAME DELINEATION ENGINE**  
**FOR OPTICAL DATA TRANSPORT**

SUBA17  
5 This application claims the benefit of U.S. Provisional  
Application No. 60/\_\_\_\_\_, filed October 29, 2001 (Express Mail,  
Label No. EL890012205US) which is hereby incorporated by reference  
in its entirety.

**Field of the Invention**

10 The present invention relates to a method and/or  
architecture for optical data transport generally and, more  
particularly, to a payload-independent frame delineation engine and  
associated protocol.

15 **Background of the Invention**

20 Information is sent over conventional optical networks as  
a series of digital bytes. There are no analog values on the  
optical line to mark start or end of a frame. With a pure stream  
of bytes, it is not possible to tell where a particular frame  
starts and ends. Frame delineation is a method to mark the  
boundaries of a frame with special patterns or parameters so that

0325.00524  
CD01153

a receiving device can separate frames within an incoming byte stream.

Conventional slow-speed wide area network (WAN) links and optical networks with Packet-over-SONET (POS) transports use a technique called High-Level Data Link Control (HDLC) to delineate frames. The HDLC method uses a unique byte value of 0x7E (hexadecimal) as a frame delimiter to mark the beginning and the end of each frame. Special coding is used inside each frame to make sure that any data pattern that matches the frame delimiter is converted to special codes to avoid false start/end indications.

Referring to FIG. 1, a drawing of a conventional HDLC encoded frame 10 is shown. Each frame 10 has a start-of-frame delimiter 12, a payload field 14, an optional cyclic redundancy check (CRC) field 16, and an end-of-frame delimiter 18. A difficulty with the HDLC approach is that there is an increased likelihood of losing the 0x7E frame delimiters 12 and 18 due to bit errors as link speeds increase. In addition to an error-prone delineation, the HDLC scheme also suffers from bandwidth inefficiency since many data byte patterns must be encoded into special two-byte sequences to avoid the 0x7E frame delimiter pattern and other control characters patterns. Therefore, the HDLC

0325.00524  
CD01153

scheme requires transmitting more bytes than the actual payload. Depending on the number of conflicting bytes inside the payload, the amount of actual bandwidth needed to send a packet can be quite large.

5                   To alleviate the limitations of HDLC, a method called Simple Data Link (SDL) was developed by Bell Labs. The SDL method does not use a predetermined pattern to delineate the start and the end of the frame. Instead, the SDL method only includes a header in each frame to delineate the start of frame. The end of frame is  
10                   calculated from the start of frame delimiter and a check for an optional payload CRC field.

                  Referring to FIG. 2, a drawing of a conventional SDL encoded frame 20 is shown. In the SDL method, a 2-byte field 22 containing a length of the payload is used as a portion of the  
15                   start-of-frame delimiter. A 2-byte length CRC field 24 that follows the length field 22 is used as a second portion of the start-of-frame delimiter. The length CRC field 24 containing a CRC value for the value stored in the length field 22. The length CRC field 24 is followed by a payload field 26 and an optional payload  
20                   CRC field 28. The value stored in the length field 22 does not include the 4-byte payload CRC field 28.

0325.00524  
CD01153

A receiving engine for SDL delineation has hunting logic that tracks an incoming frame 20 on a byte-by-byte basis to look for a pattern where a CRC computed on first two bytes (length field 22) matches the following two bytes (length CRC field 24). If a match is found, then the receiving engine marks a valid start-of-frame. A number of bytes following the length CRC field 24 are returned to protocol processing entities as the payload. The number of bytes is equal to the value in the length field 22.

Once the payload is retrieved, the receiving engine receives the 4 subsequent bytes as a payload CRC value. The payload CRC value is used to verify integrity of the payload. After the payload CRC field 28 is received, the receiving engine starts hunting for a length/CRC validation to look for next frame boundary.

A problem with the SDL approach is that many protocols such as Ethernet, frame relay, and payloads for multiprotocol over Asynchronous Transfer Mode (ATM) have their own built-in CRC as part of the payload. Therefore, the SDL frame 20 does not have the additional 4-bytes of the payload CRC field 28. Conventional multi-service transport protocols require special logic for handling specific protocols as part of the frame delineation engine

0325.00524  
CD01153

due to the optional payload CRC field 28. The special logic must understand the particular payload format being processed. Therefore, the special logic requires complicated delineation engines that need the complication of Open Systems Interconnect data link (i.e., layer 2) lookup and processing. Such a solution is complicated, expensive, and non-scalable. Each time a new protocol is added within the payload field 26, the logic must be changed to accommodate the new protocol.

For channelized SONET/SDH applications, it is almost impossible to use conventional technology for multi-service transport over different channels, due to the complexity and bulkiness of each engine. Since protocol requirements over a single channel change, each SDL engine must be able to accommodate all types of protocols. The receiving engine cannot be used to send payload data that does not have a header that clearly identifies presence or absence of the payload CRC field 28.

### Summary of the Invention

The present invention concerns a method for delineating a frame. The method generally comprises the steps of (A) receiving the frame comprising (i) a length value incorporating a payload

0325.00524  
CD01153

error detection length, (ii) a length error detection value, (iii) a payload data, and (iv) a payload error detection value having the payload error detection length, wherein the payload error detection value and the payload data occupy separate fields of the frame, (B) performing an error detection on the length value based upon the length error detection value, and (C) retrieving the payload data and the payload error detection value based upon the length value in response to passing the error detection on the length value.

The objects, features and advantages of the present invention include providing an optical data transport engine and protocol that may provide for (i) payload-independent frame delineation, (ii) payload independent frame construction, (iii) scalability, (iv) simple and efficient operation, and/or (v) cost effectiveness.

#### **Brief Description of the Drawings**

These and other objects, features and advantages of the present invention will be apparent from the following detailed description and the appended claims and drawings in which:

FIG. 1 is a drawing of a conventional HDLC encoded frame;

FIG. 2 is a drawing of a conventional SDL encoded frame;

FIG. 3 is a block diagram of an optical data transport engine;

FIG. 4 is a block diagram of an example circuit implementing framing;

5 FIG. 5 is a flow diagram of a method for framing;

FIG. 6 is a block diagram of an example circuit implementing frame delineation;

FIG. 7 is a flow diagram of a method for frame delineation; and

10 FIG. 8 is a block diagram of another embodiment of the optical data transport engine.

#### Detailed Description of the Preferred Embodiments

Referring to FIG. 3, a block diagram of an apparatus 100 is shown in accordance with a preferred embodiment of the present invention. The apparatus 100 may be implemented as an optical data transport engine. An interface 102 may be provided in the optical data transport engine 100 to receive a signal (e.g., RX\_DATA). An interface 104 may be provided in the optical data transport engine 100 to present a signal (e.g., TX\_DATA).





0325.00524  
CD01153

may connect to the interface 102 to receive the signal RX\_DATA.  
The circuit 110 may connect to the interface 104 to present the  
signal TX\_DATA.

The circuit 106 may be implemented as a receiver circuit.

5 The receiver circuit 106 may convert the signal RX\_DATA from an  
optical signal to an electrical signal for processing by the  
circuit 108. The receiver circuit 106 may perform equalization,  
amplification, clock recovery, demodulation, and/or other  
operations related to converting the signal RX\_DATA from an optical  
domain to an electrical domain.  
10

The circuit 108 may be implemented as a framing and frame  
delineation engine. The framing and frame delineation engine 108  
generally performs (i) a frame delineation operation on the  
converted signal RX\_DATA and (ii) a framing operation that  
ultimately produces the signal TX\_DATA. A particular  
15 implementation of the framing and frame delineation engine 108 may  
include one or both of the operations. Related U.S. Patent  
Application, Serial No. 09/881,493 filed on June 14, 2001, hereby  
incorporated by reference in its entirety, generally provides  
20 another example implementation for processing the frames.

The circuit 110 may be implemented as a transmitter circuit. The transmitter circuit 110 may convert an electrical signal presented by the framing and frame delineation engine 108 into an optical signal that is the signal TX\_DATA. The transmitter  
5 circuit 110 may perform source coding, channel coding, and/or other operations related to converting the signal TX\_DATA from the electrical domain to the optical domain.

104353430704  
10202F3852F00F  
The framing and frame delineation engine 108 may be intended as a payload-independent hardware engine for use primarily  
10 with optical networks. The frame delineation operation of the present invention generally involves a technique where the length field 22 (FIG. 2) includes a length of the payload field 26 and a length (e.g., 4-bytes) of the payload CRC field 28, if present. Therefore, the framing and frame delineation engine 108 may be  
15 protocol-independent, small, efficient, and cost-effective.

With Simple Data Link (SDL)-style framing gaining popularity in high-speed optical networking and standardization by the International Telecommunications Union (ITU), the American National Standards Institute (ANSI), and the Internet Engineering  
20 Task Force (IETF), the above-mentioned technique may be very useful in optical transport device designs. By changing the definition of

0325.00524  
CD01153

the length field 22 to point to an end of the frame 20 instead of an end of the payload field 26, the framing and frame delineation engine 108 may not require any additional logic to process individual protocols encapsulated by the payload field 26 and the payload CRC field 28. The encapsulated protocol may include, but is not limited to, an Ethernet (IEEE 802.3) protocol, a frame relay transport protocol, and a Generic Framing Procedure (ANSI Draft).

Referring to FIG. 4, a block diagram of an example circuit 112 implementing the framing operation is shown. The circuit 112 may be integrated into the framing and frame delineation engine 108. The circuit 112 generally comprises a circuit 114, a circuit 116, a circuit 118, a circuit 120, and a circuit 122.

The circuits 114 and 116 may receive a signal (e.g., DATA). The circuit 114 may present a signal (e.g., PAYLOAD) to the circuit 122. The circuit 114 may present another signal (e.g., PLEN) to the circuit 118. The circuit 116 may present a signal (e.g., PCRCLEN) to the circuit 118. The circuit 116 may also present another signal (e.g., PCRC) to the circuit 122. The circuit 118 may present a signal (e.g., LEN) to the circuit 120 and the circuit 122. The circuit 120 may present a signal (e.g., LCRC)

0325.00524  
CD01153

to the circuit 122. The circuit 122 may present a signal (e.g.,  
FRAME\_OUT).

The signal DATA may be implemented as a data stream of  
customer data. The signal PAYLOAD may be implemented as a buffered  
5 presentation of the signal DATA. The signal PLEN may be  
implemented as a payload length value for the signal PAYLOAD. The  
signal PCRC may be implemented as a CRC value or other error  
10 detection value calculated from the signal PAYLOAD. The signal  
PCRCLEN may be implemented as a payload CRC length value for the  
signal PCRC. The signal LEN may be implemented as a length value  
that is a sum of the payload length value and the payload CRC  
length value. The signal LCRC may be a CRC value or other error  
15 detection value calculated for the signal LEN. The signal  
FRAME\_OUT may be implemented as a modified version of the frame 20  
(e.g., having a modified definition of the length value  
incorporating the payload CRC length value).

The circuit 114 may be implemented as a data buffer  
circuit. The data buffer circuit 114 may be configured to store a  
frame worth of the customer data in the signal DATA. The circuit  
20 114 may present the customer data as the signal PAYLOAD after the  
entire frame has been buffered. The data buffer circuit 114 may

0325.00524  
CD01153

determine the payload length value from the signal PAYLOAD. The payload length value may be expressed in terms of a number of bytes. The data buffer circuit 114 may present the payload length value as the signal PLEN.

5           The circuit 116 may be implemented as a payload CRC circuit. The payload CRC circuit 116 may be configured to generate the payload CRC value based upon the signal DATA. The calculation of the payload CRC value may take place in parallel to the buffering of the signal DATA. The payload CRC circuit 116 may present the payload CRC value as the signal PCRC. The payload CRC circuit 116 may also determine the payload CRC length value for the signal PCRC. The payload CRC length value may be expressed in terms of a number of bytes. The payload CRC length value may be presented as the signal PCRCLEN.

15           The circuit 118 may be implemented as a length circuit. The length circuit 118 may be configured to calculate the length value destined for the length field 22 of the modified frame 20. The length circuit 118 may calculate the length value by summing the payload length value within the signal PLEN and the payload CRC length value within the signal PCRCLEN. The length value may be

0325.00524  
CD01153

expressed in terms of a number of bytes. The length circuit 118 may present the length value as the signal LEN.

The circuit 120 may be implemented as a length CRC circuit. The length CRC circuit 120 may be configured to calculate  
5 the length CRC value for the signal LEN. The length CRC value may be expressed in terms of a number of bytes. The length CRC circuit 120 may present the length CRC value as the signal LCRC.

The circuit 122 may be implemented as a multiplexer circuit. The multiplexer circuit 122 may be configured to  
10 multiplex the signals PAYLOAD, LEN, LCRC and PCRC to present the signal FRAME\_OUT. The multiplexer circuit 122 may multiplex in an order of (i) the signal LEN, (ii) the signal LCRC, (iii) the signal PAYLOAD, and (iv) the signal PCRC.

The modified frame 20 may be implemented without the  
15 payload CRC field 28. Where the payload CRC field 28 is not required, the payload CRC circuit 116 may set the signal PCRCLEN to a zero value and set the signal PCRC to a null value. Furthermore, the multiplexer 122 may skip multiplexing the signal PCRC into the signal FRAME\_OUT. Therefore, the signal LEN may be set to the  
20 length of just the payload.

0325.00524  
CD01153

Referring to FIG. 5, a flow diagram for a framing operation of the circuit 112 is shown. The framing operation may begin with the buffering of the signal DATA (e.g., block 124) by the data buffer circuit 114. The circuit 116 may calculate the payload CRC value (e.g., block 126). The payload CRC value calculation may be performed simultaneously with the buffering operation. Once all of the data has been buffered and the payload CRC value calculated, the length circuit 118 may determine the length value from the signal PLEN and the signal PCRCLEN (e.g., block 128). The length circuit 120 may then calculate the length CRC value from the signal LEN (e.g., block 130).

The multiplexer circuit 122 may assemble the signal FRAME\_OUT (e.g., block 132). Frame assembly may begin by multiplexing the signal LEN to the signal FRAME\_OUT (e.g., block 134). The signal LCRC may then be multiplexed to the signal FRAME\_OUT (e.g., block 136). The signal PAYLOAD may then be multiplexed to the signal FRAME\_OUT (e.g., block 138). Finally, the signal PCRC may be multiplexed to the signal FRAME\_OUT (e.g., block 140). The resulting signal FRAME\_OUT may be presented to the transmitter circuit 110 for transmission as the signal TX\_DATA.

0325.00524  
CD01153

Referring to FIG. 6, a block diagram of an example circuit 142 implementing the frame delineation operation is shown. The circuit 142 may be integrated into the framing and frame delineation engine 108. The circuit 142 generally comprises a circuit 144, a circuit 146, a circuit 148, a circuit 150, and a circuit 152. The circuits 144 and 152 may receive a signal (e.g., FRAME\_IN). The circuit 144 may present a signal (e.g., GROUP\_1) to the circuit 146. The circuit 144 may also present a signal (e.g., GROUP\_2) to the circuit 148. The circuit 146 may present the signal LCRC to the circuit 148. The circuit 146 may also present the signal LEN to the circuit 150. The circuit 148 may present a signal (e.g., PASS) to the circuit 146 and external to the circuit 142. The circuit 150 may present a signal (e.g., SEL) to the circuit 152. The circuit 152 may present the signal DATA. The circuit 152 may also present the signal PCRC.

The signal FRAME\_IN may be implemented as a modified frame 20 received by the receiver circuit 106. The signal FRAME\_IN may be an electrical domain version of the signal RX\_DATA. The signal GROUP\_1 may be implemented as a predetermined number of bytes sampled from the signal FRAME\_IN. The signal GROUP\_2 may be implemented as another predetermined number of bytes sampled from



0325.00524  
CD01153

the signal FRAME\_IN. The signal PASS may be implemented as a pass/fail indicator. The signal SEL may operate as a demultiplex selection signal.

The circuit 144 may be implemented as a buffer circuit.

5 The buffer circuit 144 may be configured to buffer several bytes of data from the signal FRAME\_IN. The buffer circuit 144 may buffer the several bytes of data in a first-in-first-out operation.

10 A predetermined number of oldest bytes (e.g., 2 bytes) buffered from the signal FRAME\_IN may be presented as the signal GROUP\_1. Another predetermined number of bytes (e.g., 2 bytes) immediately following the predetermined number of oldest bytes may be presented as the signal GROUP\_2. The signals GROUP\_1 and GROUP\_2 may be used by the circuit 146 and the circuit 148 to search for the length field 22 and the length CRC field 24 of the modified frame 20.

20 The circuit 146 may be implemented as another length CRC circuit. The length CRC circuit 146 may be configured to calculate the length CRC value from the bytes presented in the signal GROUP\_1. The length CRC circuit 146 may also buffer the bytes presented in the signal GROUP\_1 internally.

0325.00524  
CD01153

The circuit 148 may be implemented as a compare circuit. The compare circuit 148 may be configured to compare the signal LCRC with the signal GROUP\_2. The compare circuit 148 may assert the signal PASS when the signals LCRC and GROUP\_2 match to mark a start of the modified frame 20. The compare circuit 148 may de-assert the signal PASS when the signals LCRC and the GROUP\_2 do not match. The length CRC circuit 146 may present the buffered signal GROUP\_1 as the signal LEN in response to the signal PASS being asserted. The length CRC circuit 146 may not present the signal LEN in response to the signal PASS being de-asserted.

The circuit 152 may be implemented as a demultiplexer circuit. The demultiplexer circuit 150 may demultiplex or block the signal FRAME\_IN as determined by the signal SEL. While demultiplexing, the signal FRAME\_IN may be presented as the signal DATA or the signal PCRC as selected by the signal SEL. While blocking, the signal FRAME\_IN may not be presented at a used output of the demultiplexer circuit 152.

The circuit 150 may be implemented as a counter circuit. The counter circuit 150 may use the signal LEN to count bytes within the signal FRAME\_IN. When the counter circuit 150 receives the signal LEN from the length CRC circuit 146, the counter circuit

0325.00524  
CD01153

150 may present the signal SEL with a value that may cause the demultiplexer circuit 152 to route the signal FRAME\_IN to the signal DATA. As each byte of the signal FRAME\_IN is demultiplexed, the counter circuit 150 may count down to a predetermined value equal to the length of the payload CRC (e.g., 4 bytes if the payload CRC is present and zero bytes if the payload CRC is absent). When the counter circuit 150 reaches the predetermined value, the value of the signal SEL may be changed to cause the demultiplexer circuit 152 to route the signal FRAME\_IN to the signal PCRC, if present. The counter circuit 150 may continue counting bytes of the signal FRAME\_OUT until the count value reaches zero. Thereafter, the counter circuit 150 may present a value in the signal SEL that may cause the demultiplexer circuit 152 to block the signal FRAME\_OUT.

Referring to FIG. 7, a flow diagram for a frame delineation operation of the circuit 142 is shown. The frame delineation operation generally (i) receives the length field 22, (ii) validates the length field 22 with length CRC field 24, (iii) delineates the payload field 26 and payload CRC field 28 based upon the value given in the length field 22, and then (iv) begins searching for the next frame. The frame delineation operation may

0325.00524  
CD01153

begin by buffering a portion of the signal FRAME\_IN (e.g., block 154) in the buffer circuit 144. Two bytes from the buffer circuit 144 may be presented as the signal GROUP\_1 (e.g., block 156). The length CRC circuit 146 may generate a CRC value for the two bytes read from the signal GROUP\_1 (e.g., block 158). The compare circuit 148 may then compare the CRC value for the two bytes against the next two bytes presented by the buffer circuit 144 as the signal GROUP\_2 (e.g., decision block 160).

If the CRC value of the first two bytes does not match the value of the second two bytes (e.g., the NO branch of decision block 160), then the buffer circuit 144 may advance the buffered data by a byte sampled from the signal FRAME\_IN (e.g., block 162). The process then generally iterates at reading the two oldest bytes as the signal GROUP\_1 (e.g., block 156). The iterations may continue through consecutive bytes of the signal FRAME\_IN until the CRC value calculated from the signal GROUP\_1 matches the value from the signal GROUP\_2 (e.g., the YES branch of decision block 160) thus marking a start to the modified frame 20. The demultiplexer circuit 152 may then retrieve and separate the payload data and the payload CRC value from the signal FRAME\_IN under the control of the counter circuit 150 (e.g., block 164). In one embodiment, the

0325.00524  
CD01153

process may use the length value captured and presented by the length CRC circuit 146 to jump ahead the length value number of bytes in the signal FRAME\_IN to being searching for a next frame (e.g., block 166).

5 Referring to FIG. 8, a block diagram of another example apparatus 100A implementing the optical data transport engine is shown. At sufficiently low data rates for the signals RX\_DATA and TX\_DATA, the frame delineation operation and the framing operation may be implemented under software control. Therefore, the optical data transport engine 100A may comprise a computer circuit 166 and a storage medium 168. The storage medium 168 may store a software program 170 that is readable and executable by the computer circuit 166. The software program 170 may contain instructions implementing the frame delineation operation and/or the framing operation.

10 The function performed by the flow diagrams of FIGS. 5 and 7 may be implemented using a conventional general purpose digital computer programmed according to the teachings of the present specification, as will be apparent to those skilled in the relevant art(s). Appropriate software coding can readily be prepared by skilled programmers based on the teachings of the

0325.00524  
CD01153

present disclosure, as will also be apparent to those skilled in the relevant art(s).

The present invention may also be implemented by the preparation of ASICs, FPGAs, or by interconnecting an appropriate network of conventional component circuits, as is described herein, modifications of which will be readily apparent to those skilled in the art(s).

The present invention thus may also include a computer product which may be a storage medium including instructions which can be used to program a computer to perform a process in accordance with the present invention. The storage medium can include, but is not limited to, any type of disk including floppy disk, optical disk, CD-ROM, and magneto-optical disks, ROMs, RAMs, EPROMs, EEPROMs, Flash memory, magnetic or optical cards, or any type of media suitable for storing electronic instructions.

As used herein, the term "simultaneously" is meant to describe events that share some common time period but the term is not meant to be limited to events that begin at the same point in time, end at the same point in time, or have the same duration.

While the invention has been particularly shown and described with reference to the preferred embodiments thereof, it

0325.00524  
CD01153

will be understood by those skilled in the art that various changes in form and details may be made without departing from the spirit and scope of the invention.

10047528 100704  
10047528 100704